

020

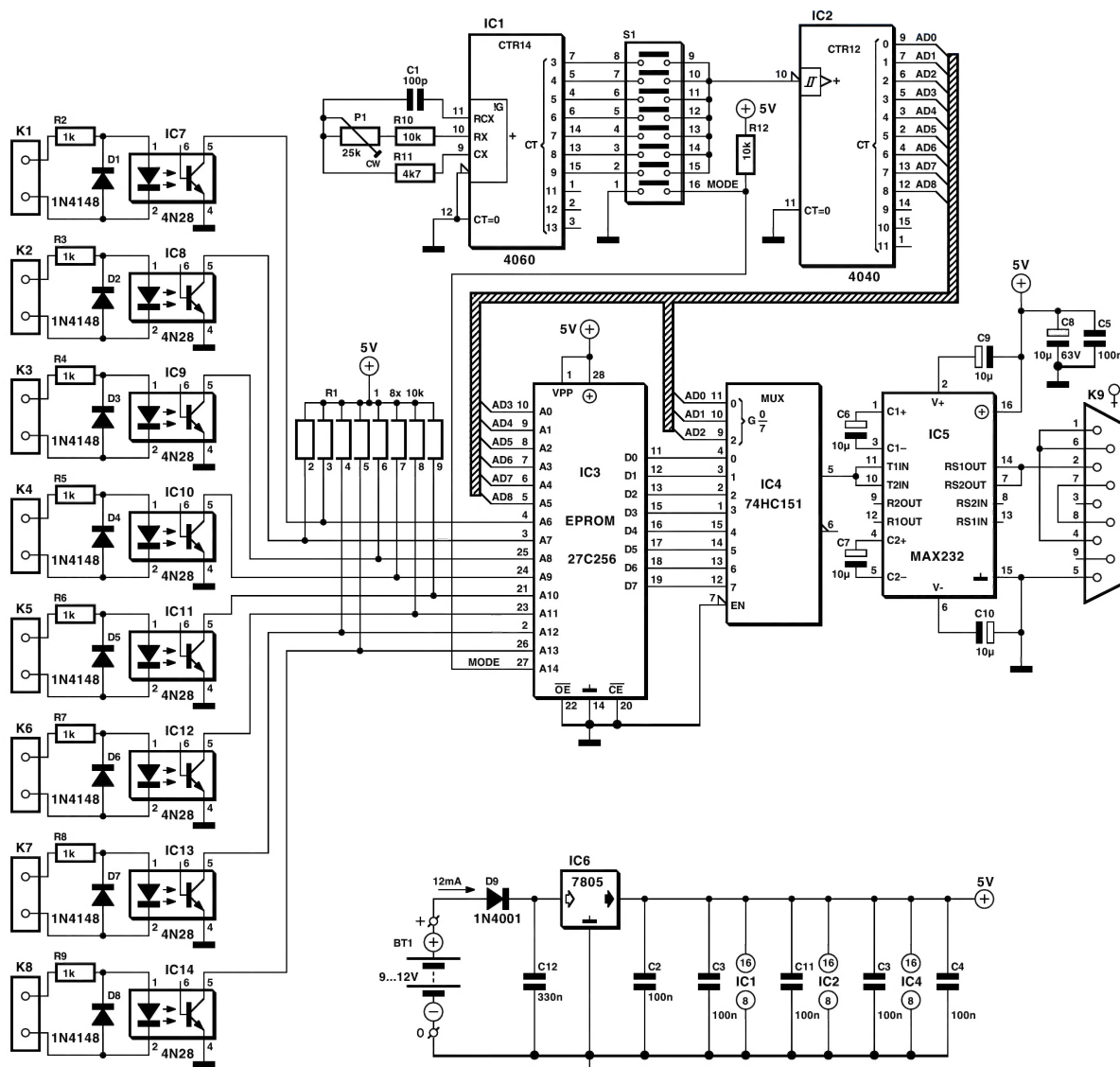
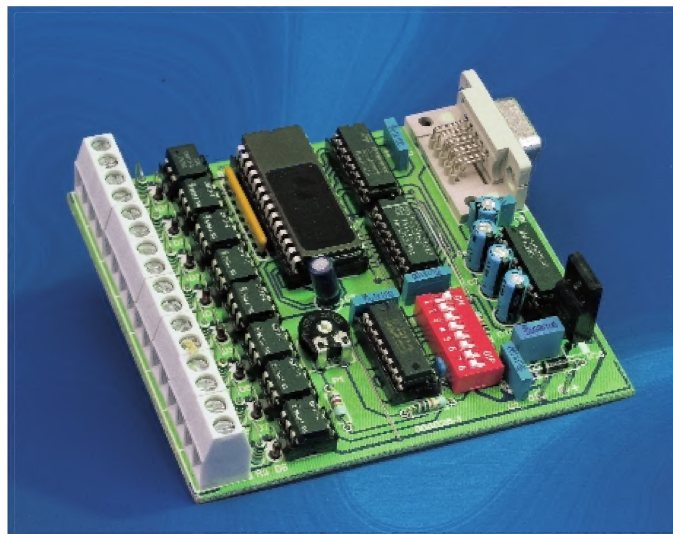
8 Channel D/I Card for RS232

G. Vastianos

The author is a student at the Electronics Department, Technological Educational Institute of Piraeus, Greece.

This project is a card with eight optically isolated TTL-compatible inputs for external connection to the serial port, which communicates through the RS-232 protocol. On this card you will look in vain for a UART or microcontroller chip because the design is based on a 27C256 EPROM.

Each digital input is connected to an optocoupler (IC7-IC14) through a resistor (R2-R9), and drives it with a current of 5 mA when the input voltage is 5 Vdc. A diode (D1-D8) is connected in parallel with each optocoupler input to protect against the ill effects of reversed signal polarity. The collectors of the internal optocoupler transistors drive EPROM address lines A6 to A13. The logic states of A6 to A13 are inverted (because the optocouplers work as inverters) and for this reason a second inversion is performed in software (EPROM) to cancel the total inver-



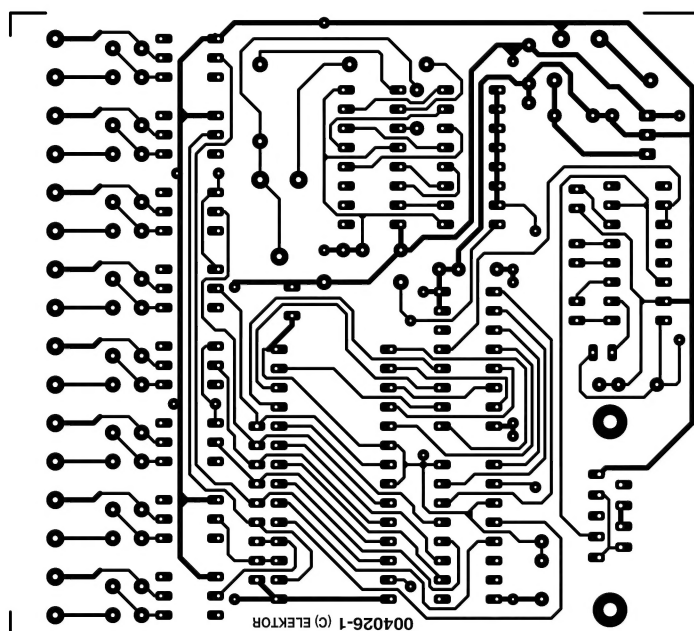
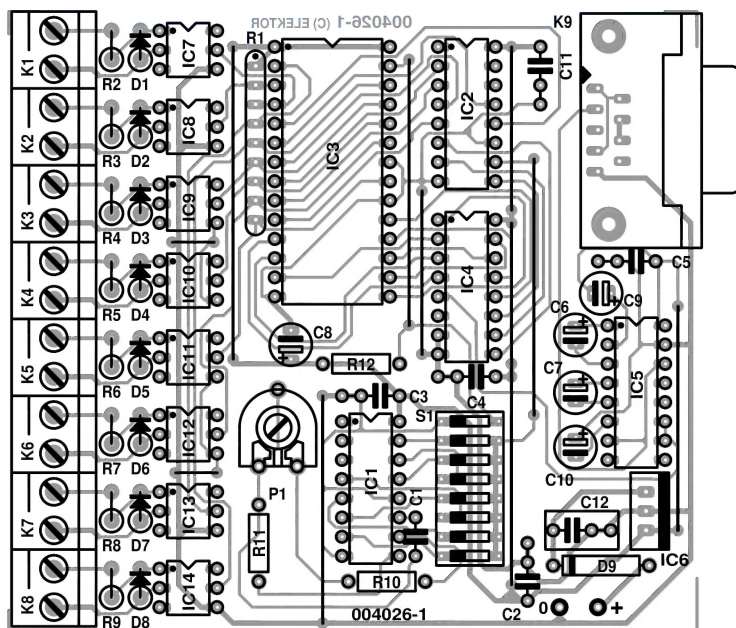


Table 1. ASCII Codes & bit values

CHAR	ASCII	D7	D6	D5	D4	D3	D2	D1	D0
CR	13	0	0	0	0	1	1	0	1
LF	10	0	0	0	0	1	0	1	0
'C'	67	0	1	0	0	0	0	1	1
'H'	72	0	1	0	0	1	0	0	0
'O'	48	0	0	1	1	0	0	0	0
'1'	49	0	0	1	1	0	0	0	1
'2'	50	0	0	1	1	0	0	1	0
'3'	51	0	0	1	1	0	0	1	1
'4'	52	0	0	1	1	0	1	0	0
'5'	53	0	0	1	1	0	1	0	1
'6'	54	0	0	1	1	0	1	1	0
'7'	55	0	0	1	1	0	1	1	1
','	58	0	0	1	1	1	0	1	0
SP	32	0	0	1	0	0	0	0	0
SYNC	255	1	1	1	1	1	1	1	1

COMPONENTS LIST

Resistors:

R1 = 10k Ω 8-way SIL array

$$R2-R9 = 1k\Omega$$
$$R_{10}, R_{12} = 10k\Omega$$
 $R_{11} = 4k\Omega$

P1 = 25k Ω preset H

Capacitors:

C1 = 100pF

C2-C5,C11 = 100nF

C6-C10 = 10 μ F radial

$$C_{12} = 330\text{nF}$$

Semiconductors:

D1-D8 = 1N4148

D9 = 1N4001

IC1 = 4060

IC2 = 4040

IC3 = 27C256 (see text for programming)

IC4 = 74HC151

$$IC5 = MAX232$$
$$IC6 = 7805$$

IC7-IC14 = 4N28 of CNY17-2

Miscellaneous:

K1-K8 = 2-way PCB terminal block, raster 5mm

K9 = 9-way sub-D socket (female), PCB mount

S1 = 8-way DIP switch

sion effect. In this way the 8 inputs become TTL compatible.

IC1, a CD4060, with the help of C1, P1, R11 and R10, forms a bitrate generator for 150, 300, 600, 1200, 2400, 4800 or 9600 bps. Preset P1 needs to be adjusted to the desired baudrate — an accuracy of 1-2% is required! The bitrate is selected using DIP switch S1. The CD4040 works as a 9-bit binary counter. The three LS bits of the counter outputs (AD0, AD1, AD2) drive the selection inputs of multiplexer IC4 (74HC151). The other counter bits (AD3-AD8) drive the LS address bits of the EPROM (A0-A5). The highest EPROM address line, A14, is connected to one of the DIP switches in S1 which defines the data transmission mode (TTY or BIN).

With the above connections counter IC2 sweeps a range of 512 bits (64 bytes) of the total of 256 kBits (32 Kbytes) contained in the EPROM.

The number of 512-bit blocks (packets) equals the number of different combinations of CH0 to CH7 and MODE, and can be calculated from the following formula :

$$\begin{aligned} \text{Memory}_{\text{size}} &= 512 \text{ packets} * 512 \text{ bits/packet} \\ &= 256 \text{ Kbits} = 32 \text{ Kbytes} \end{aligned}$$

The data encoding unit consisting of IC1- IC4 transmits 512-bits packets (coming from the multiplexer output). The data in the transmitted packet depends directly on the logic states of lines CH0 to CH7 and MODE.

The power supply includes a 7805 regulator so that the card will not need a separate regulated power supply. Circuit IC5 (a MAX232) works as an RS232 Driver/Receiver, converting the multiplexer's output data from TTL to RS232 compatible.

With the EPROM programmed with the right data, the circuit will produce messages (packets), through the RS-232 protocol, that will inform us for the logic states of the card's inputs. The transmission used by this card is **8 data bits, 1 stop bit and no parity**. Assuming an 'A' is to be sent, and that your PC runs a terminal simulation program receiving (via the RXD pin of its serial port) the following 10 bits '0100000101', then it will print on its screen the character 'A'.

This card transmits the logic states of its inputs in one of two modes, TTY (teletype) or BIN (binary).

In **TTY Mode**, the card transmits the following message to the computer:

```
<SYNC>,<SYNC>,<CR>,<LF>,'CH0:X',<SP>,'CH1:X',
<SP>,'CH2:X',<SP>,'CH3:X',<SP>,'CH4:X',<SP>,'CH5:X',<SP>,'CH6:X',<SP>,'CH7:X'
```

Where <SYNC> is the character with ASCII code 255, used to synchronize the computer. Reception of this character (from the computer) causes a space to appear on the screen. <CR> and <LF> are 'carriage return' and 'line feed' with ASCII codes 13 and 10 respectively, and <SP> is the 'space' character with ASCII code 32. Reception of this character (from the computer) causes a space to appear on the screen. Finally, X is the logic state of each input, which will cause a '1' or a '0' to appear on the screen.

In **BIN Mode**, the card transmits the following message to the computer:

```
<SYNC>,<SYNC>,<DATA BYTE>
```

Where <SYNC> is as above and <DATA BYTE> is the character with an ASCII code equal to the numerical value of the byte built from the logic states of the inputs (with the MS Bit

representing CH7, and LS Bit, CH0).

These two modes are used in different cases. In cases where we want just to see the logic states of the input we just set the card in TTY mode and we use a terminal program as Telix, Procomm Plus, Hyper Terminal etc. In cases where we want to do some data logging or to compose a data acquisition & control system with other DAQ & control cards then we just set the card in BIN mode (because the decoding of the data is easier done by dedicated software).

The characters, their ASCII codes and their bit values, as used in the two modes are shown in **Table 1**. The bitstreams that must be transmitted for each character appear in **Table 2**.

The contents of the EPROM are created by a program (EPROMFMP.BAS) developed in Quick Basic, which is available from the author's website. The program first makes a temporary file that contains all data to be written in the EPROM in bit format (08DICARD.TMP). Next, it converts this file into byte format (08DICARD.BIN) and deletes the temporary file. Finally, it runs an external program (BIN2HEX.EXE) and adapts the file from Binary to IntelHex format (08DICARD.HEX). To program your own EPROM, you may use 08DICARD.BIN or 08DICARD.HEX file.

Finally, the mode and bit rate selection on S1 is shown in **Table 3**. The PCB designed for this project is unfortunately not available ready-made through the Publishers' Readers Services.

All software you will need to complete this project may be downloaded free of charge from the author's website at <http://members.xoom.com/robofreak/download/08dicard.htm>. Also, for other projects developed by Mr. Vastianos, visit this address:

<http://members.xoom.com/robofreak/>

(004026-1)

Table 2. Character composition

CHAR	START BIT	D0	D1	D2	D3	D4	D5	D6	D7	STOP BIT
CR	0	1	0	1	1	0	0	0	0	1
LF	0	0	1	0	1	0	0	0	0	1
'C'	0	1	1	0	0	0	0	1	0	1
'H'	0	0	0	0	1	0	0	1	0	1
'0'	0	0	0	0	0	1	1	0	0	1
'1'	0	1	0	0	0	1	1	0	0	1
'2'	0	0	1	0	0	1	1	0	0	1
'3'	0	1	1	0	0	1	1	0	0	1
'4'	0	0	0	1	0	1	1	0	0	1
'5'	0	1	0	1	0	1	1	0	0	1
'6'	0	0	1	1	0	1	1	0	0	1
'7'	0	1	1	1	0	1	1	0	0	1
':'	0	0	1	0	1	1	1	0	0	1
SP	0	0	0	0	0	0	1	0	0	1
SYNC	0	1	1	1	1	1	1	1	1	1

Table 3. DIP switch settings

SW								1	USAGE
#1	#2	#3	#4	#5	#6	#7	#8		MODE
OFF	ON	OFF	OFF	OFF	OFF	OFF	OFF	OFF	"150, 8, 1, N : BIN"
OFF	OFF	ON	OFF	OFF	OFF	OFF	OFF	OFF	"300, 8, 1, N : BIN"
OFF	OFF	OFF	ON	OFF	OFF	OFF	OFF	OFF	"600, 8, 1, N : BIN"
OFF	OFF	OFF	OFF	ON	OFF	OFF	OFF	OFF	"1200, 8, 1, N : BIN"
OFF	OFF	OFF	OFF	OFF	ON	OFF	OFF	OFF	"2400, 8, 1, N : BIN"
OFF	OFF	OFF	OFF	OFF	OFF	ON	OFF	OFF	"4800, 8, 1, N : BIN"
OFF	OFF	OFF	OFF	OFF	OFF	OFF	ON	OFF	"9600, 8, 1, N : BIN"
ON	ON	OFF	OFF	OFF	OFF	OFF	OFF	OFF	"150, 8, 1, N : TTY"
ON	OFF	ON	OFF	OFF	OFF	OFF	OFF	OFF	"300, 8, 1, N : TTY"
ON	OFF	OFF	ON	OFF	OFF	OFF	OFF	OFF	"600, 8, 1, N : TTY"
ON	OFF	OFF	OFF	ON	OFF	OFF	OFF	OFF	"1200, 8, 1, N : TTY"
ON	OFF	OFF	OFF	OFF	ON	OFF	OFF	OFF	"2400, 8, 1, N : TTY"
ON	OFF	OFF	OFF	OFF	OFF	ON	OFF	OFF	"4800, 8, 1, N : TTY"
ON	OFF	OFF	OFF	OFF	OFF	OFF	ON	OFF	"9600, 8, 1, N : TTY"